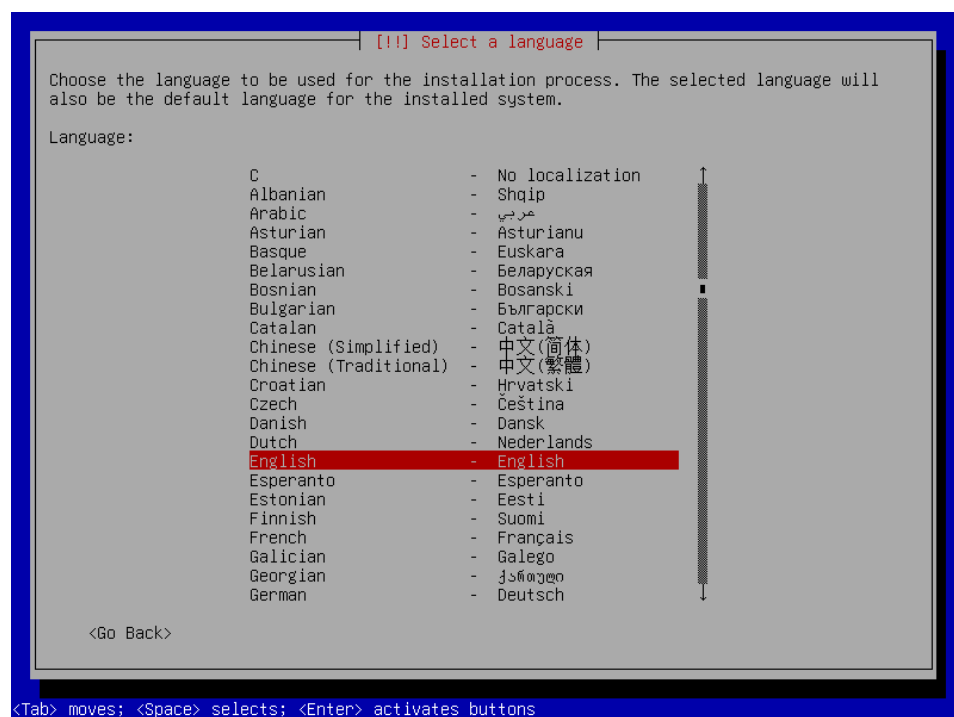
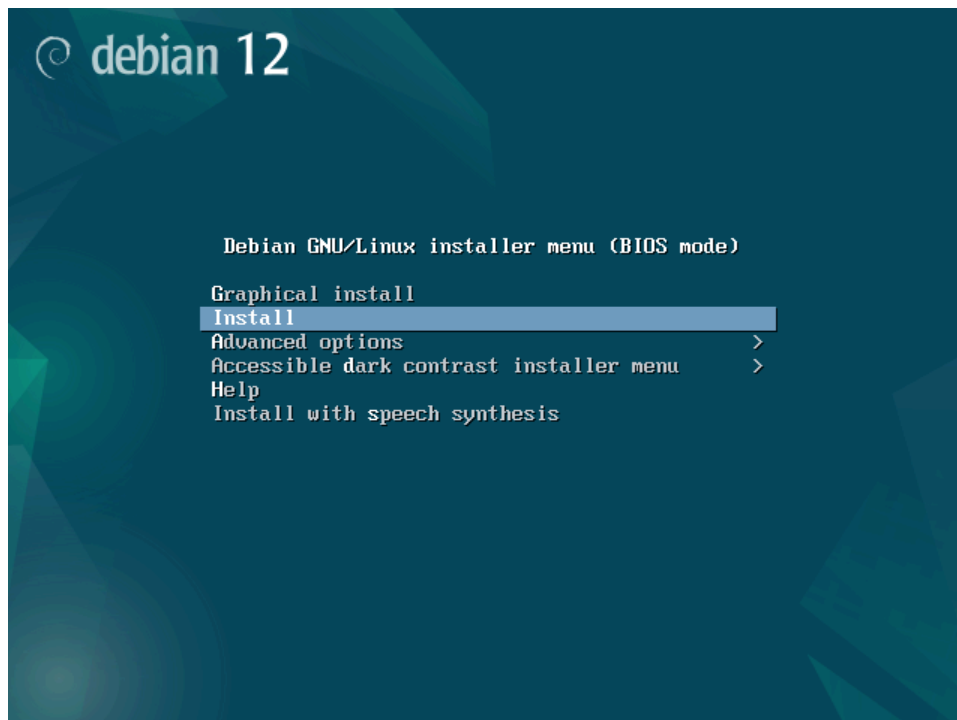


# MISE EN PLACE D'UN SERVEUR NAS

## I – MISE EN PLACE D'UNE VM DEBIAN (SANS GUI)



[!] Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

nas-server

<Go Back>

<Continue>

[!!] Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

Root password:

\*\*\*\*\*

[ ] Show Password in Clear

<Go Back>

<Continue>

[!!] Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

Allaoui

<Go Back>

<Continue>

[!!] Set up users and passwords

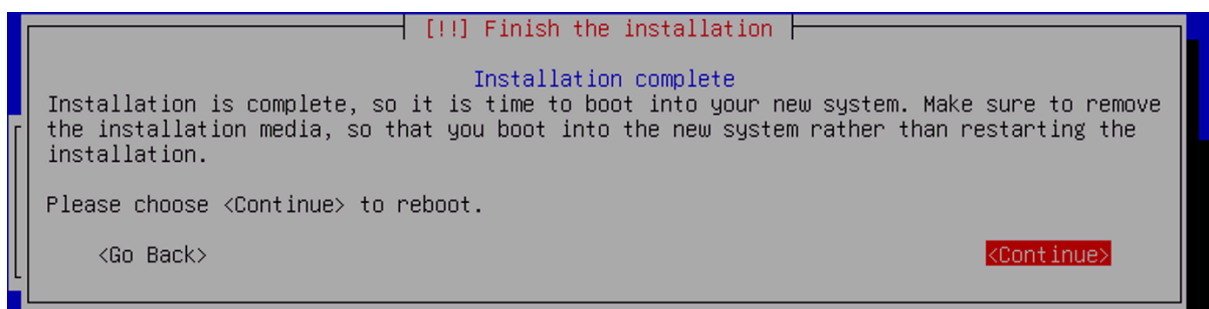
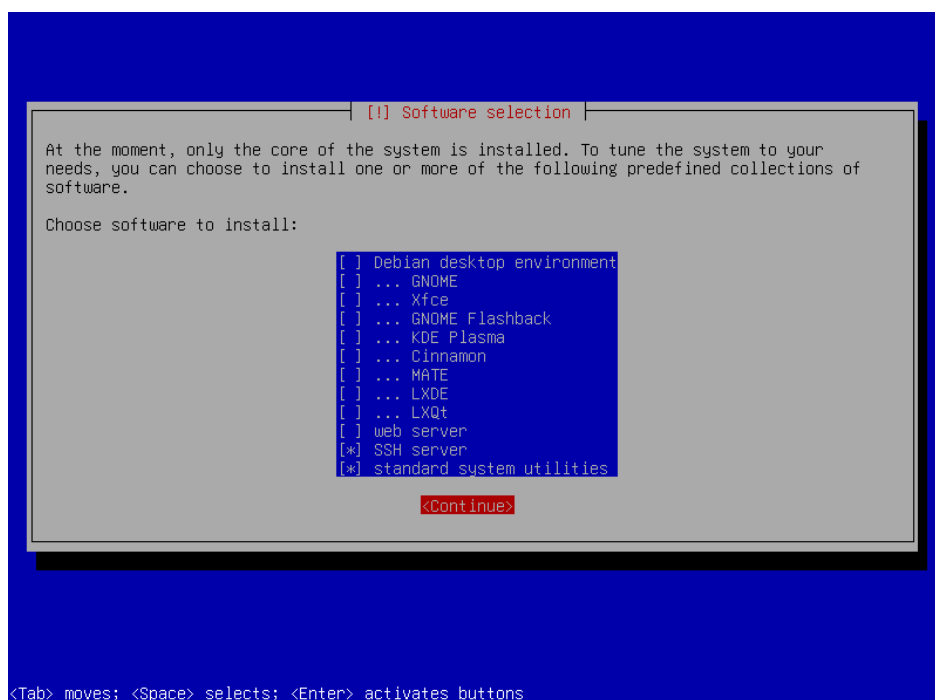
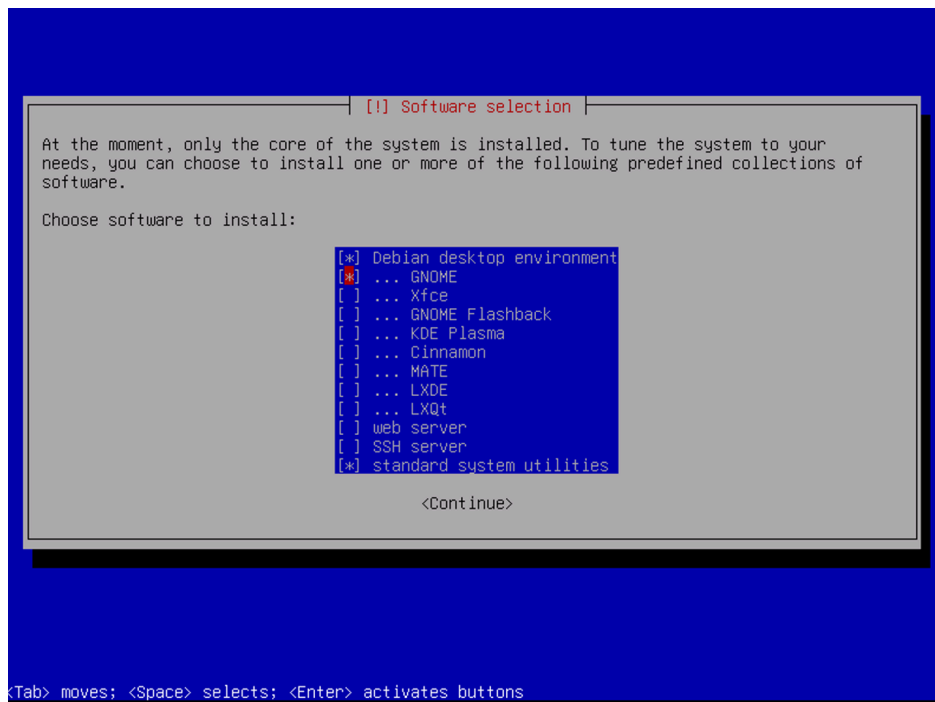
Select a username for the new account. Your first name is a reasonable choice. The username should start with a lower-case letter, which can be followed by any combination of numbers and more lower-case letters.

Username for your account:

allaoui

<Go Back>

<Continue>



## II – PRÉPARATION DE L'ENVIRONNEMENT

### Fixer une adresse IP au serveur

Premièrement on va attribuer une adresse IP statique à notre serveur, pour faire cela nous allons chercher à connaître notre interface réseau avec la commande :

*ip a* ou *ip addr*

```
root@nas-server:/# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:2b:43:e7 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.18/24 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 969sec preferred_lft 969sec
    inet6 fe80::20c:29ff:fe2b:43e7/64 scope link
        valid_lft forever preferred_lft forever
root@nas-server:/#
```

Maintenant nous allons chercher aussi à savoir notre nameserver

Pour connaître notre nameserver, on va utiliser la commande :

*cat /etc/resolv.conf*

```
root@nas-server:/# cat /etc/resolv.conf
domain localdomain
search localdomain
nameserver 192.168.1.2
root@nas-server:/#
```

Ensuite, avec nano on va accéder à **/etc/network/interfaces** et appliquez les configuration nécessaires :

- Identifier votre interface réseau
- Remplacez "inet **dhcp**" par "**static**".
- Ensuite l'attribuer l'adresse **ip** que l'on a choisie, ainsi que son **masque réseau** , **passerelle**, etc..."

```
# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 192.168.1.18
netmask 255.255.255.0
gateway 192.168.1.2
dns-nameservers 192.168.1.2
broadcast 192.168.1.255
```

### III – CONFIGURATION DU RAID

#### Élévation des droits d'un utilisateur

Pour configurer notre **RAID**, nous aurons besoin d'un utilisateur **non-root** disposant des privilèges **sudo**.

Mais avant tout ne pas oublier d'installer le paquet sudo en utilisant la commande suivante (en étant en root) : **apt-get install sudo**

Pour accorder les privilèges **sudo** à un utilisateur, il est nécessaire de l'ajouter au groupe **sudo** en utilisant la commande suivante : **sudo usermod -aG sudo utilisateur**

#### Détails des commandes :

**usermod** : La commande de base utilisée pour modifier un compte utilisateur.

**-aG** : L'option **-a** (append) ajoute l'utilisateur au (**G**) groupe spécifié sans le retirer des autres groupes.

#### Installation du RAID

Pour notre serveur, nous allons configurer un système **RAID** de **niveau 5 (RAID 5)**, ce qui nécessitera au moins trois disques durs.

*Avant et après ajout des disques :*

```
allaoui@nas-server:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   10G  0 disk
├─sda1       8:1    0    9G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0   975M  0 part [SWAP]
sr0         11:0    1 1024M  0 rom
```

```
allaoui@nas-server:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   10G  0 disk
├─sda1       8:1    0    9G  0 part /
├─sda2       8:2    0    1K  0 part
└─sda5       8:5    0   975M  0 part [SWAP]
sdb          8:16    0    2G  0 disk
sdc          8:32    0    2G  0 disk
sdd          8:48    0    2G  0 disk
sr0         11:0    1 627M  0 rom
```

Nous allons maintenant installer la dépendance "**mdadm**" nécessaire à la création du RAID.

***sudo apt install mdadm -y***

```
allaoui@nas-server:~$ sudo apt install mdadm -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mdadm is already the newest version (4.2-5).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

On identifie les noms des nouveaux disques durs ajoutés (**sdb**, **sd****c**, **sdd**) pour ensuite configurer notre RAID avec la commande suivante :

***sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd***

```
allaoui@nas-server:~$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
[sudo] password for allaoui:
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
allaoui@nas-server:~$
```

### Détails des commandes :

1. **mdadm** : C'est l'outil utilisé pour gérer les ensembles RAID dans Linux.
2. **--create /dev/md0** : Indique à **mdadm** de créer un nouvel ensemble RAID qu'on va appelé **/dev/md0**. Le nom **md0** est un nom de périphérique RAID typique.
3. **--level=5** : le niveau RAID à utiliser.
4. **--raid-devices=3** : Indique que l'ensemble RAID sera composé de 3 périphériques de stockage.
5. **/dev/sdb /dev/sdc /dev/sdd** : Les trois périphériques de stockage (disques durs) qui seront utilisés pour créer l'ensemble RAID.

## Créez le fichier de configuration pour conserver la configuration RAID

**sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf**

```
allaoui@nas-server:~$ sudo mdadm --detail --scan | sudo tee -a /etc/mdadm/mdadm.conf
ARRAY /dev/md0 metadata=1.2 name=nas-server:0 UUID=66fdc85b:e5d7b481:221e21f2:b9476ba4
allaoui@nas-server:~$
```

### Détails des commandes :

1. **sudo mdadm --detail --scan** : Cette commande scanne tous les ensembles RAID actifs sur le système et affiche leurs informations détaillées.
2. **|** : (Pipe) Redirige la sortie de la commande précédente (**mdadm --detail --scan**) vers l'entrée de la commande suivante (**tee**).
3. La commande **tee** va lire et écrire le résultat de la commande précédente(1). combiner avec l'option **-a** signifie "append" (ajouter à la fin du fichier). Donc, ajoutez les informations scannées à la fin du fichier du fichier de configuration /etc/mdadm/mdadm.conf

**But :** Mettre à jour le fichier de configuration **mdadm.conf** avec les informations des ensembles RAID actuellement actifs. Cela garantit que les configurations RAID sont persistantes et reconnues au redémarrage du système.

Ensuite la commande : **sudo update-initramfs -u**

```
allaoui@nas-server:~$ sudo update-initramfs -u
[sudo] password for allaoui:
update-initramfs: Generating /boot/initrd.img-6.1.0-21-amd64
```

1. Cette commande met à jour l'image **initramfs**.
2. Initramfs (**initial RAM filesystem**) est une image initiale utilisée pour monter temporairement le système de **fichiers root** avant que le vrai système de fichiers ne soit monté.
3. L'option **-u** signifie "update" (mettre à jour).

**But :** Après avoir modifié la configuration RAID, il est nécessaire de mettre à jour **l'initramfs** pour s'assurer que les nouvelles configurations RAID sont prises en compte lors du démarrage du système.

Cela permet au système d'initialiser correctement les ensembles RAID lors du démarrage.

### Formatez et montez le RAID

Maintenant, nous allons créer un système de fichiers de type ext4 et le monter sur notre RAID avec les commandes suivantes :

1. Formatez l'ensemble RAID **/dev/md0** avec le système de fichiers ext4 :  
`sudo mkfs.ext4 /dev/md0`
2. Créez un répertoire pour servir de point de montage :  
`sudo mkdir -p /mnt/nas`
3. Montez le système de fichiers du RAID sur le répertoire nouvellement créé :  
`sudo mount /dev/md0 /mnt/nas`

```
allaoui@nas-server:~$ cd /mnt/nas
allaoui@nas-server:/mnt/nas$ ls
lost+found
allaoui@nas-server:/mnt/nas$
```

Ensuite va on l'ajouter au **fstab** pour le monter automatique au démarrage avec la commande : **echo '/dev/md0 /mnt/nas ext4 defaults 0 0' | sudo tee -a /etc/fstab**

Le fichier **fstab (File System Table)** (situé dans **/etc/fstab**) est le fichier de configuration qui contient les informations sur le montage des systèmes de fichiers. Il liste tous les disques et partitions disponibles, et indique où les monter dans l'arborescence du système Linux, avec quelles options. (source : [linuxtrix.fr](http://linuxtrix.fr))



## IV – CONFIGURATION DES SERVICES DE FICHIERS (WEBDAV, SFTP, SAMBA)

### Installation et configuration d'Apache et WebDAV

Installer Apache et les utilitaires : **sudo apt install apache2 apache2-utils -y**

```
root@nas-server:/home/allaoui# sudo apt install apache2 apache2-utils -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
apache2 is already the newest version (2.4.59-1~deb12u1).
apache2-utils is already the newest version (2.4.59-1~deb12u1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@nas-server:/home/allaoui#
```

Activation des modules nécessaires : **sudo a2enmod dav dav\_fs auth\_digest**

```
allaoui@nas-server:~$ sudo a2enmod dav dav_fs auth_digest
Enabling module dav.
Considering dependency dav for dav_fs:
Module dav already enabled
Enabling module dav_fs.
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
    systemctl restart apache2
allaoui@nas-server:~$
```

Modifier le fichier de configuration d'Apache :

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

----- > Fichier conf ici < -----

Créer les répertoires partagés : **sudo mkdir -p /mnt/nas/partage/public**

Créer les sous-dossiers dans le répertoire public : **cd /mnt/nas/partage/public**

**sudo mkdir DOCUMENTS\_RH PHOTOS PROJETS RESSOURCES VIDEOS**

```
allaoui@nas-server:/mnt/nas/partage/public$ ls
DOCUMENTS_RH  PHOTOS  PROJETS  RESSOURCES  VIDEOS
```

Créer les répertoires utilisateurs :

```
sudo mkdir -p /mnt/nas/partage/users/elias
```

```
sudo chown root:root /mnt/nas/partage/users/elias
```

 <= À répéter pour chaque users

```
sudo chmod -R 770 /mnt/nas/partage/users/elias
```

```
sudo chown elias:elias /mnt/nas/partage/users/elias
```

```
allaoui@nas-server:/mnt/nas/partage/users$ ls -l
total 12
drwxrwx--- 3 allaoui allaoui 4096 May 21 18:45 allaoui
drwxrwx--- 3 elias   elias   4096 May 21 18:44 elias
drwxrwx--- 3 rachid  rachid  4096 May 21 18:43 rachid
```

Attribuer les droits aux répertoires :

```
sudo chown -R www-data:www-data /mnt/nas/partage/public
```

```
sudo chmod -R 775 /mnt/nas/partage/public
```

```
allaoui@nas-server:/mnt/nas/partage$ ls -l
total 8
drwxrwxr-x 8 www-data www-data 4096 May 21 21:46 public
drwxr-xr-x 5 root      root    4096 May 18 17:57 users
allaoui@nas-server:/mnt/nas/partage$
```

Ensuite, vous ajoutez les utilisateurs au groupe `www-data` pour qu'ils puissent lire et écrire avec la commande suivante : `sudo usermod -aG www-data elias`

Et vice-versa, vous ajoutez `www-data` au groupe de chaque utilisateur avec cette commande : `sudo usermod -aG elias www-data`

```
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG www-data elias
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG www-data allaoui
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG www-data rachid
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG rachid www-data
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG allaoui www-data
allaoui@nas-server:/mnt/nas/partage/users$ sudo usermod -aG elias www-data
```

Activer le site par défaut et redémarrer Apache :

```
sudo a2ensite 000-default.conf
```

```
sudo systemctl restart apache2
```

Créer le fichier pour stocker les utilisateurs et mots de passe :

```
sudo touch /etc/apache2/users.password
```

Attribuer les droits appropriés au fichier :

```
sudo chown www-data:www-data /etc/apache2/users.password
```

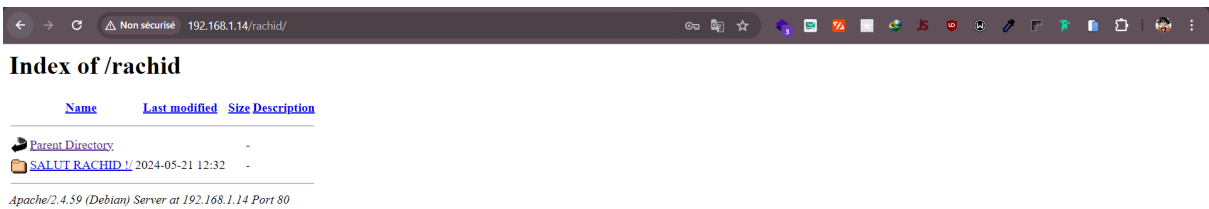
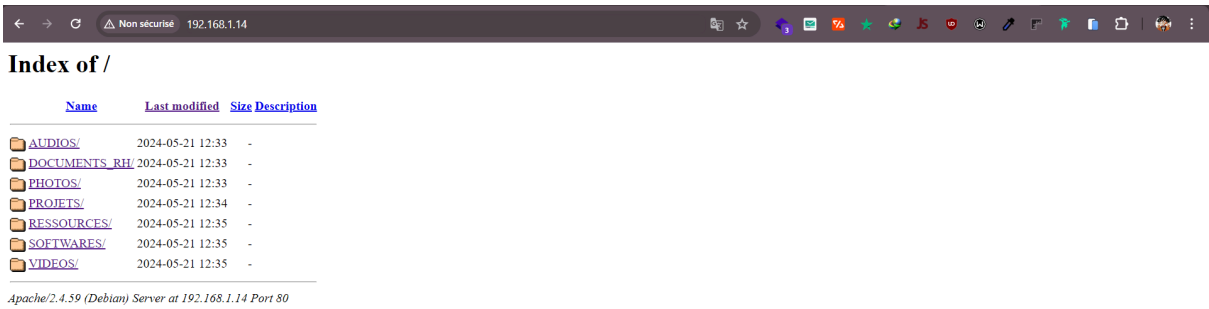
Créer des utilisateurs avec leurs mots de passe :

```
sudo htdigest /etc/apache2/users.password allaoui allaoui
```

```
sudo htdigest /etc/apache2/users.password elias elias
```

```
sudo htdigest /etc/apache2/users.password rachid rachid
```

Redémarrer Apache : `sudo systemctl restart apache2`



## Sécuriser le serveur avec un certificat SSL

Tout d'abord, installer OpenSSL en utilisant la commande suivante :

```
sudo apt install openssl
```

```

allaoui@nas-server:~$ sudo apt install openssl
[sudo] password for allaoui:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.11-1~deb12u2).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
allaoui@nas-server:~$

```

Ensuite, générez un certificat auto-signé avec cette commande :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/leserveur-nas.com.key -out /etc/ssl/certs/leserveur-nas.com.crt
```

[illegible]

Maintenant faut accédez au répertoire **/etc/apache2/sites-available/** et modifiez le fichier **default-ssl.conf** de notre serveur :

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Ajoutez ces lignes au fichier **default-ssl.conf** :

*SSLEngine on*

*SSLCertificateFile /etc/ssl/certs/leserveur-nas.com.crt*

*SSLCertificateKeyFile /etc/ssl/private/leserveur-nas.com.key*

----- > Fichiers default-ssl.conf < -----

Activer SSL : **sudo a2enmod ssl**

Ensuite, on va rediriger toutes les requêtes HTTP vers HTTPS en ouvrant le fichier **000-default.conf** : **sudo nano /etc/apache2/sites-available/000-default.conf**

Ajoutez cette ligne dans le fichier :

**Redirect permanent / https://192.168.1.14 // IP de votre serveur**

```
<VirtualHost *:80>
    ServerAdmin webmaster@leserveur-nas.com
    DocumentRoot /mnt/nas/partage/public
    ServerName 192.168.1.14
    Redirect permanent / https://192.168.1.14




    <Directory /mnt/nas/partage/public>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
```

Ensuite on redémarre apache et on vérifie : **sudo systemctl restart apache2**

← ↻

Not secure | https://192.168.1.14

# Index of /

Name	Last modified	Size	Description
 <a href="#">DOCUMENTS_RH/</a>	2024-05-21 18:32	-	
 <a href="#">PHOTOS/</a>	2024-05-21 18:32	-	
 <a href="#">PROJETS/</a>	2024-05-21 18:32	-	
 <a href="#">READ_ME.txt</a>	2024-05-21 18:45	29	
 <a href="#">RESSOURCES/</a>	2024-05-21 18:32	-	
 <a href="#">VIDEOS/</a>	2024-05-21 18:32	-	

Apache/2.4.59 (Debian) Server at 192.168.1.14 Port 443

## Installez et configurez OpenSSH pour SFTP

Dans cette section, nous allons mettre en place et configurer **SFTP**.

Pour le faire, nous commencerons par installer **OpenSSH** en utilisant la commande suivante : ***sudo apt-get install openssh-server***

Après l'avoir installer on va ensuite modifier le fichier de configuration **SSH** :

***sudo nano /etc/ssh/sshd\_config***

```
Include /etc/ssh/sshd_config.d/*.conf

Port 2468
PermitEmptyPasswords no
PasswordAuthentication yes
PermitRootLogin no

# override default of no subsystems
#Subsystem      sftp    /usr/lib/openssh/sftp-server
Subsystem       sftp    internal-sftp

# Example of overriding settings on a per-user basis
Match group public
    X11Forwarding no
    AllowTcpForwarding no
    ChrootDirectory /mnt/nas/partage
    ForceCommand internal-sftp
```

***sudo systemctl restart sshd***

----- >Fichier sshd\_config [ici](#) < -----

## TEST DE CONNEXION !!!

On peut voir sur la première image que j'essaie de me connecter en **SSH**, mais que je rencontre une erreur.

La raison est simple : l'option « **p** » en minuscule est utilisée pour **SSH**, tandis que la majuscule « **P** » est réservée pour **SFTP**.

```
C:\Users\Ah La Oui>ssh -P 2468 allaoui@192.168.1.14
ssh: connect to host 2468 port 22: Unknown error

C:\Users\Ah La Oui>ssh -p 2468 allaoui@192.168.1.14
allaoui@192.168.1.14's password:
Last login: Sat May 18 17:53:31 2024 from 192.168.1.1

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
allaoui@nas-server:~$
```

```
C:\Users\Ah La Oui>sftp -p 2468 elias@192.168.1.14
ssh: connect to host 2468 port 22: Unknown error
Connection closed

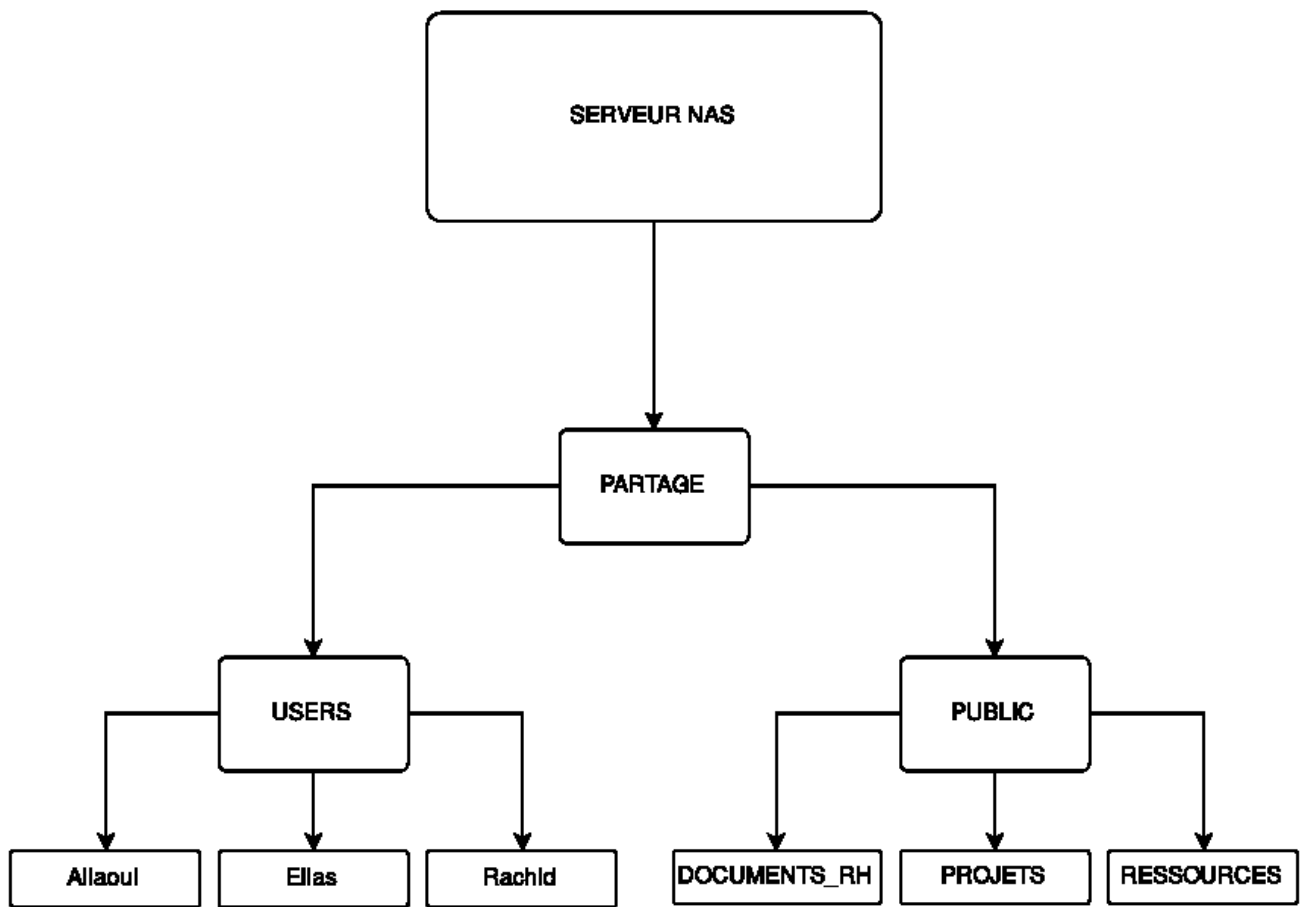
C:\Users\Ah La Oui>sftp -P 2468 elias@192.168.1.14
elias@192.168.1.14's password:
Connected to 192.168.1.14.
sftp> ls
users
sftp>
```

-p : **Utiliser en SSH** (Minuscule)

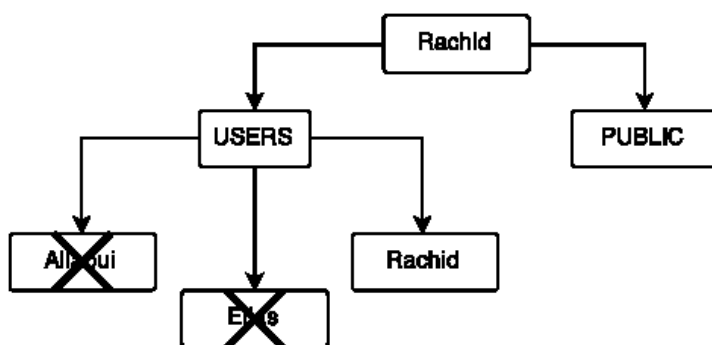
-P : **en SFTP** (Majuscule)

---

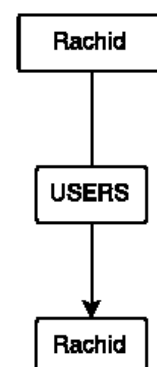
## PETIT VISUEL DE L'ENSEMBLE DE NOTRE PARTAGE



### EXEMPLE D'UNE CONNNEXION D'UN USERS EN SFTP



### EXEMPLE D'UNE CONNNEXION EN WEBDAV





## 1. Configuration du partage de dossiers/fichiers avec SAMBA

### Installation de SAMBA

**Note** : les commandes sont exécutées directement avec l'utilisateur "root" mais si vous agissez depuis un compte administrateur (autre que root), pensez à préfixer les commandes avec "sudo".

Mettre les paquets à jour sur notre machine Linux :

```
root@nas-server:~# sudo apt-get update
```

Ensuite, installer le paquet **"samba"**

```
root@nas-server:~# sudo apt-get install samba
```

Cette commande installera et démarrera à la fois le serveur **Samba smbd** et le serveur Samba NetBIOS **nmbd**. **nmbd** n'est pas obligatoire pour ce tutoriel, donc pour des raisons de sécurité vous pouvez l'arrêter et le désactiver avec systemctl:

```
root@nas-server:~# sudo systemctl stop nmbd.service
```

```
root@nas-server:~# sudo systemctl disable nmbd.service
```

La `sudo systemctl disable nmbd.service` commande produira le résultat suivant lors de son exécution :

```
root@nas-server:~# Output
nmbd.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install disable nmbd
insserv: warning: current start runlevel(s) (empty) of script 'nmbd' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script 'nmbd' overrides LSB defaults (0 1 6).
```

Cette sortie indique que, comme **nmbd** elle n'a pas **systemd** de configuration de gestion native, elle est désactivée par l'ancien système d'initialisation SysV.

Pour éviter les problèmes de sécurité pouvant survenir lors de l'exécution d'un service réseau non configuré, arrêtons le serveur Samba jusqu'à ce que les détails de configuration soient en place :

```
root@nas-server:~# sudo systemctl stop smbd.service
```

Suite à l'installation, on peut afficher la version installé de samba par commande suivante :

```
root@nas-server:~# smbld --version
```

Pour afficher le statut de "samba", pour voir s'il est activé ou arrêté, voici la commande :

```
root@nas-server:~# systemctl status smbld
```

Le résultat :

```
root@debian:~# systemctl status smbld
● smbld.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbld.service; enabled; preset: enabled)
   Active: active (running) since Thu 2024-05-16 11:22:23 CEST; 2h 8min ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 3219 ExecCondition=/usr/share/samba/is-configured smb (code=exited, status=0/SUCCESS)
  Process: 3221 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 3225 (smbd)
    Status: "smbd: ready to serve connections..."
     Tasks: 4 (limit: 2264)
    Memory: 7.2M
       CPU: 2.265s
    CGroup: /system.slice/smbld.service
            └─3225 /usr/sbin/smbd --foreground --no-process-group
              └─3227 /usr/sbin/smbd --foreground --no-process-group
                └─3228 /usr/sbin/smbd --foreground --no-process-group
                  └─3335 /usr/sbin/smbd --foreground --no-process-group
```

Maintenant, passons à la création du partage de dossier

## Création de notre premier partage sous Samba

La création du partage va s'effectuer en plusieurs étapes : la configuration de Samba dans un premier temps, et la préparation du groupe, de l'utilisateur et du dossier du partage dans un second temps.

### 1.1. Configurer le partage dans smb.conf

Plutôt que de le modifier `/etc/samba/smb.conf` directement, renommez-le `smb.conf.original` et créez un nouveau fichier portant le nom `smb.conf` :

```
root@nas-server:~# sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig
```

Le fichier de configuration de Samba est "`/etc/samba/smb.conf`", nous allons l'éditer :

```
root@nas-server:~# sudo nano /etc/samba/smb.conf
```

La [global]section de ce fichier définit le nom du serveur, son rôle et d'autres détails, y compris les interfaces réseau :

```
GNU nano 7.2 /etc/samba/smb.conf
[global]
    server string = samba_server
    server role = standalone server
    interfaces = ens33
    bind interfaces only = yes
    disable netbios = yes
    smb ports = 445
    log file = /var/log/samba/smb.log
    max log size = 10000
```

Ces directives précisent les éléments suivants :

**server string**– Il s'agit des informations d'identification qui seront fournies aux utilisateurs lors des connexions. Vous pouvez utiliser `samba_server` ou un autre nom qui identifiera votre serveur. Tout au long de ce didacticiel, vous verrez la chaîne `samba.example.com` désignant le partage Samba pour l'organisation [Laplateforme.com](https://laplateforme.com) .

**server role**– Ceci définit quel type de serveur Samba sera créé. Dans ce cas, il s'agit d'un **standalone server**, c'est-à-dire d'un partage de fichiers. Les autres types de serveurs incluent les serveurs membres du domaine et les contrôleurs de domaine.

**interfaces**– Ce sont les interfaces réseau auxquelles Samba se liera. Ici c'est l'interface de bouclage (127.0.0.1) et est obligatoire. Vous devrez également inclure l'interface réseau externe que vous avez générée précédemment. C'est généralement **eth0**.

**bind interfaces only**– Cela garantit que Samba se lie uniquement aux interfaces répertoriées sur la interfaces ligne. Par mesure de sécurité, Samba ignore les paquets qui ne correspondent pas au fichier interfaces.

**disable netbios**– Cela désactive toutes les fonctions NetBIOS qui ne sont pas nécessaires sur un serveur autonome. Cela simplifie le processus de résolution de nom de serveur et le transport du trafic SMB.

**smb ports** – Ceci définit le port sur lequel Samba écoutera. Port 445 est le port standard de Samba.

**log file** – Ceci définit le nom et l'emplacement du fichier journal de Samba.

**max log size** – Cela définit une limite de taille sur le fichier journal. Le nombre indiqué est en octets et équivaut à 10 Mo. Quelques points à garder à l'esprit lors de la définition de cette limite de taille : Lorsqu'elle est atteinte, Samba générera un nouveau fichier journal et déplacera l'ancien contenu vers un duplicata avec une .old extension. Si la limite est à nouveau dépassée, le .old fichier existant sera détruit. Cela évite que l'espace disque/partition ne soit submergé par le contenu d'un seul fichier journal. Vous devez donc définir une taille de fichier adaptée aux ressources de votre système.

Si vous souhaitez une journalisation plus détaillée pendant la configuration du serveur, ajoutez la ligne suivante à la **[global]** section :

## 1.2. Créer un utilisateur et le groupe

Dans cette étape, nous allons créer des utilisateurs pouvant accéder aux partages. Ils auront besoin d'un accès en tant qu'utilisateurs Samba et système afin de s'authentifier auprès du serveur Samba lorsqu'ils se connecteront et liront et écriront sur le système de fichiers.

Dans l'entreprise hypothétique [Laplateforme.com](https://laplateforme.com), trois employés doivent être ajoutés au serveur Samba et créés en tant qu'utilisateurs sur le système Linux : Rachid, Elias et Alaoui. En plus de ces quatre, il y aura un utilisateur administrateur qui pourra accéder et administrer les partages personnels. Cet utilisateur sera également propriétaire des actions communes auxquelles tout le monde pourra accéder.

La première étape pour ajouter des utilisateurs système consiste à créer des répertoires personnels pour chacun d'eux. Plutôt que d'utiliser les

répertoires personnels standard à l'adresse **/home/user**, les répertoires et les données Samba seront situés à l'adresse **/mnt/md0/samba/**. Conserver les données Samba dans un emplacement unique et les séparer des autres données utilisateur faciliterait les tâches de gestion futures telles que les sauvegardes.

**Remarque** : Les utilisateurs créés dans ce guide ne sont pas destinés à disposer de connexions SSH. Si vos utilisateurs ont déjà des comptes sur le serveur, vous devez leur créer un utilisateur Samba dédié afin de suivre ce guide.

La section suivante expliquera le processus d'ajout du premier utilisateur, Rachid , mais vous devrez répéter ce processus pour Elias et Alaoui . La première étape consiste à créer le répertoire où seront stockées les données Samba, à la racine du système de fichiers. Ce répertoire sera appelé **/mnt/md0/samba/** et sa propriété de groupe sera définie sur **sambashare**, un groupe créé lors de l'installation de Samba.

Exécutez les commandes suivantes pour créer le **/mnt/md0/samba/** répertoire et définir la propriété du groupe sur **sambashare** :

On va créer notre répertoire en utilisant notre RAID 5 précédemment créé :

```
root@nas-server:~# sudo mkdir /mnt/md0/samba/
```

```
root@nas-server:~# sudo chown :sambashare /mnt/md0/samba/
```

Ensuite, créez le répertoire personnel de Rachid sous le **/mnt/md0/samba/** répertoire :

```
root@nas-server:~# sudo mkdir /mnt/md0/samba/rachid/
```

```
root@nas-server:~# sudo adduser --home mnt/md0/samba/rachid --no-create-home --shell /usr/sbin/nologin --ingroup sambashare rachid
```

Les options effectuent les opérations suivantes :

**--home** - Ceci définit l'emplacement du répertoire personnel de l'utilisateur.

**--no-create-home** – Cela empêche la commande adduser de créer le répertoire personnel de Rachid . Si le système devait créer ce répertoire, il serait rempli de fichiers de configuration tels que

**.bash\_history** ceux qui ne sont pas nécessaires à la configuration actuelle.

**--shell** – Ceci définit quel shell Rachid se verra attribuer lorsqu'il se connectera par SSH. Une connexion SSH n'est pas nécessaire pour accéder à un partage Samba ; définir cette valeur sur

**/usr/sbin/nologin** désactivera les connexions SSH.

**--in-group sambashare** – Cela ajoute l'utilisateur au groupe **sambashare**, lui donnant un accès en lecture et en écriture à ses propres partages et au partage commun.

Un mot de passe vous sera demandé lorsque vous exécutez cette commande. Choisissez un mot de passe unique, non basé sur un dictionnaire, de 10 caractères ou plus.

Maintenant que l'utilisateur système Rachid existe, vous pouvez définir la propriété et les autorisations sur son répertoire personnel Samba :

```
root@nas-server:~# sudo chown rachid:sambashare /mnt/md0/samba/rachid/
```

```
root@nas-server:~# sudo chmod 2770 /mnt/md0/samba/rachid/
```

Définir les autorisations du répertoire sur 2770 signifie que les nouveaux fichiers ou répertoires créés sous **/mnt/md0/samba/rachid/** hériteront de la propriété du groupe du répertoire parent plutôt que du groupe principal de l'utilisateur qui a créé le fichier ou le répertoire. Cela signifie, par exemple, que si l'utilisateur admin créait un nouveau répertoire dans le partage de Rachid , Rachid pourrait y lire et y écrire.

Ensuite, ajoutez Rachid au serveur Samba. Samba conserve sa propre base de données d'utilisateurs et de mots de passe, qu'il utilise pour authentifier les connexions. Pour se connecter, tous les utilisateurs doivent être ajoutés au serveur Samba et activés. Exécutez les commandes suivantes **smbpasswd** pour accomplir ces deux tâches :

```
root@nas-server:~# sudo smbpasswd -a rachid
```

```
root@nas-server:~# sudo smbpasswd -e rachid
```

Les options utilisées ici effectuent les opérations suivantes :

**-a** - Cela ajoute l'utilisateur au serveur Samba sans l'activer.

**-e** - Cela active un utilisateur précédemment ajouté.

Le mot de passe que vous entrez ici sera utilisé pour accéder au partage Samba et peut différer du mot de passe système.

L'utilisateur Rachid existe désormais en tant qu'utilisateur système sans possibilité de se connecter en SSH au serveur. Il possède un répertoire personnel sur **/mnt/md0/samba/rachid/**, et est enregistré et activé en tant qu'utilisateur Samba.

Répétez ce processus pour chaque utilisateur de Samba ( Elias et Alaoui etc... )

Pour créer l' utilisateur administrateur , exécutez les commandes suivantes, en changeant le répertoire personnel en **/mnt/md0/samba/everyone/** qui représentera notre dossier public auquel tous les utilisateurs peuvent accéder. Pour ce faire, il faut répéter les mêmes étapes que pour l'utilisateur Rachid.

En plus de créer l' utilisateur admin , créons un groupe appelé admins pour faciliter la gestion du serveur. Avec des autorisations de lecture et d'écriture sur chaque partage, ce groupe peut simplifier le travail d'ajout

et de suppression d'utilisateurs. Par exemple, si des utilisateurs individuels fonctionnent en tant qu'utilisateurs administrateurs puis quittent l'organisation, ils doivent être supprimés individuellement de chaque partage. Les nouveaux administrateurs doivent également être ajoutés manuellement à chaque partage. Créer un groupe d'administrateurs et accorder à ce groupe un accès en lecture-écriture aux partages signifie que l'ajout et la suppression d'utilisateurs ne nécessitent qu'une seule commande.

Exécutez les commandes suivantes pour créer un nouveau groupe appelé **admins** et ajoutez l'utilisateur **admin** à ce groupe :

```
root@nas-server:~# sudo groupadd admins
```

```
root@nas-server:~# sudo usermod -G admins admin
```

Des utilisateurs supplémentaires peuvent être ajoutés au groupe d'administrateurs en exécutant la deuxième commande et en remplaçant **.sudo usermod -G admins admin admin**

Les configurations du système sont maintenant terminées, avec les utilisateurs de l'organisation [Laplateforme.com](https://laplateforme.com) définis comme utilisateurs système et Samba. Passons à la configuration du serveur Samba afin que ces utilisateurs puissent accéder à leurs répertoires de partage.

Chaque partage aura sa propre section dans le fichier de configuration principal de Samba, **/etc/samba/smb.conf** suivant les paramètres globaux. Ces sections définissent le fonctionnement de chaque partage.

Utilisez **nano** à nouveau l'éditeur de texte pour ouvrir et modifier ce fichier :



```
GNU nano 7.2 /etc/samba/smb.conf
[rachid]
    path = /mnt/md0/samba/rachid
    browseable = yes
    read only = no
    force create mode = 0660
    force directory mode = 2770
    valid users = rachid @admins
```

Ces options incluent :

**share\_name** – C'est le nom du partage que vous utiliserez lors de votre connexion.

**path** – Il s'agit du chemin absolu du partage dans le système de fichiers.

**browsable** – Ceci définit si les autres utilisateurs peuvent ou non voir le partage. L'activation de cette option permet uniquement aux autres utilisateurs du serveur Samba de voir l'existence du partage. Il ne confère aucune autorisation de lecture ou d'écriture.

**read only** – Ceci définit si les utilisateurs **valid users** peuvent écrire sur le partage.

**force create mode** – Cela force les autorisations pour tout fichier écrit sur le partage.

**force directory mode** – Cela force les autorisations pour tout répertoire créé dans le partage.

**valid users** – Il s'agit d'une liste des utilisateurs qui ont accès au partage. Ce paramètre peut prendre des noms d'utilisateur ou des groupes système tels que admins . Les groupes doivent être répertoriés avec un **@** devant, par exemple **@admins**.

Créez un bloc de partage pour Elias et Alaoui et d'autres utilisateurs selon vos besoins . Modifiez uniquement le partage **[name]**, pathet **valid users** pour refléter chacun des noms d'utilisateur.

Le **[everyone]** partage sera différent des autres en termes de **[name]**, **path**, **valid users** et **browsable** d'options, et ressemblera à ceci :

```
GNU nano 7.2 /etc/samba/smb.conf *

[alaoui]
    path = /mnt/md0/samba/alaoui
    browseable = no
    read only = no
    force create mode = 0660
    force directory mode = 2770
    valid users = alaoui @admins

[everyone]
    path = /mnt/md0/samba/everyone
    browseable = yes
    read only = no
    force create mode = 0660
    force directory mode = 2770
    valid users = @smbashare @admins
```

Donner au groupe **smbashare** un accès en lecture-écriture au partage permet à tous les utilisateurs d'accéder au partage, puisqu'ils ont été ajoutés à ce groupe lors de leur création.

Une fois la configuration terminée, démarrons le serveur Samba avec **systemctl** :

```
root@nas-server:~# sudo systemctl start smbd.service
```

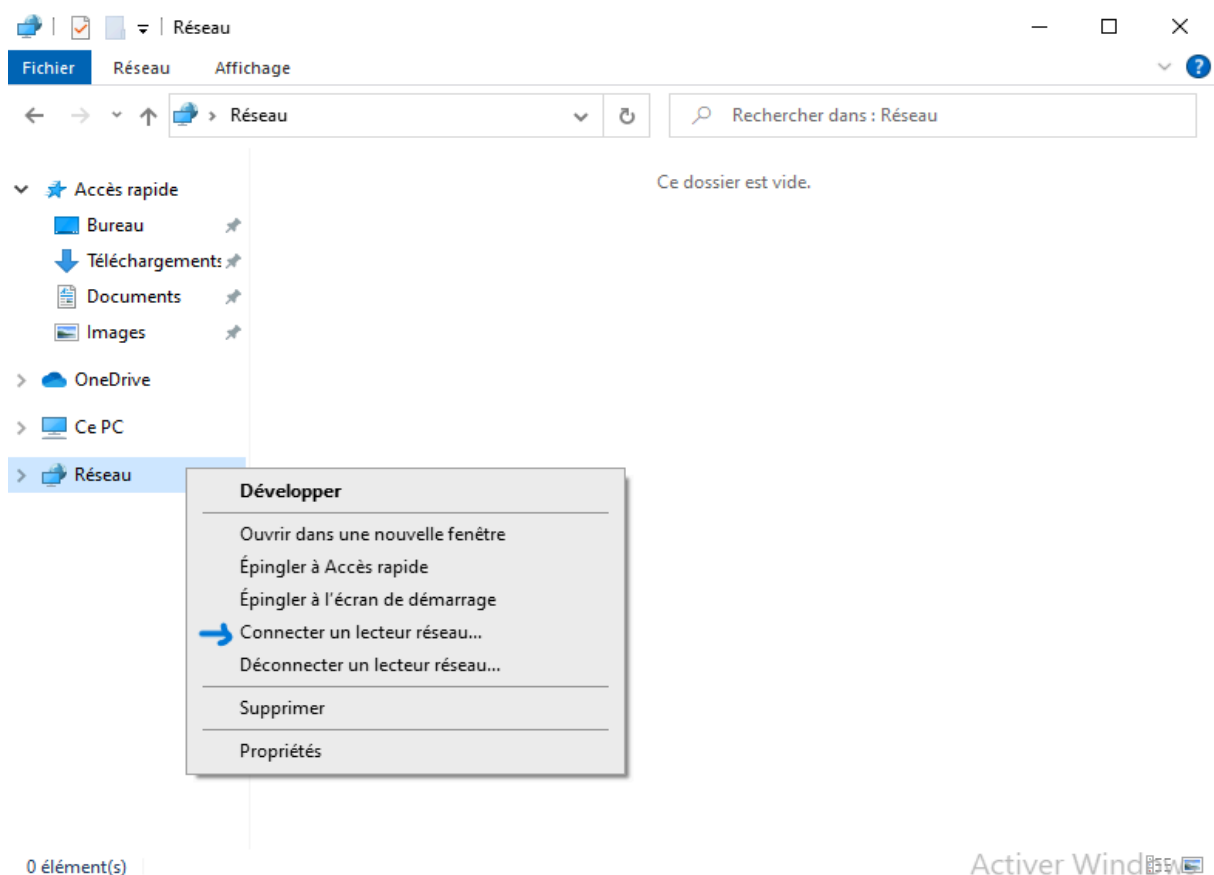
Le Serveur Samba est maintenant opérationnel

### 1.3. Connexion au serveur Samba

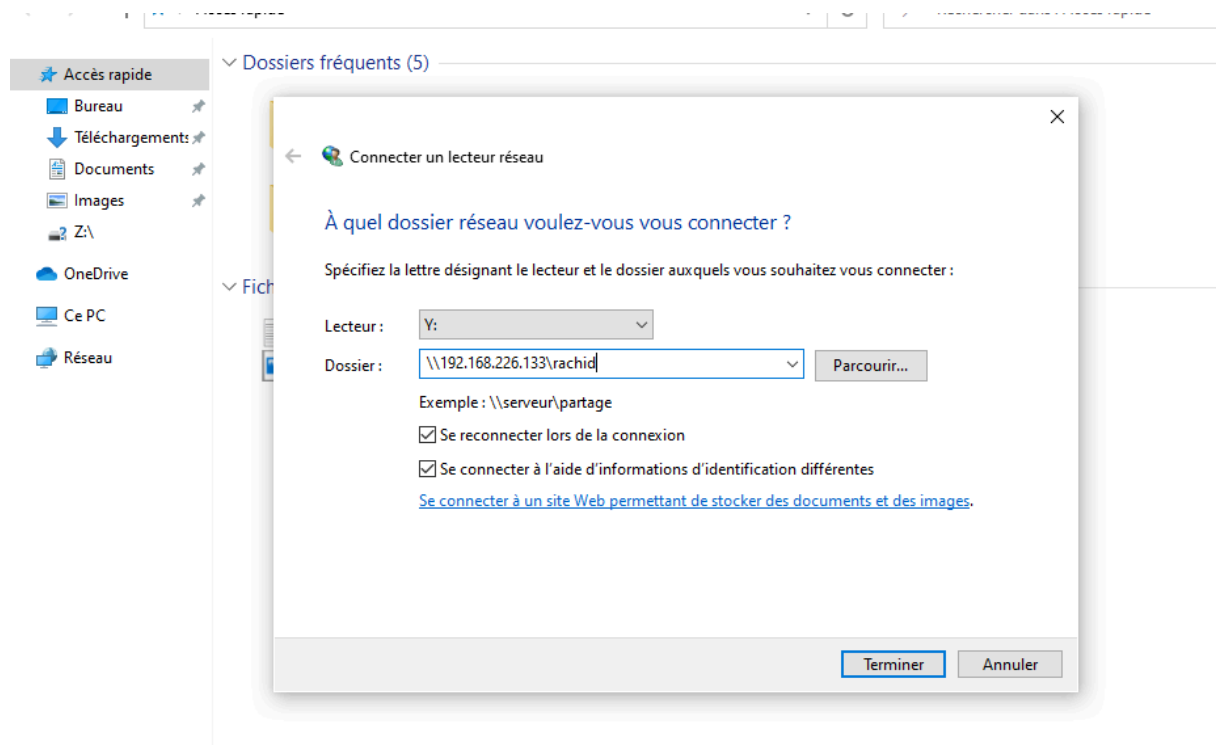
**Tout est prêt, nous pouvons tester depuis un poste client Windows ou Linux, mais ne vous inquiétez pas, nous ferons tous les tests possibles.**

#### Accéder au partage Samba depuis Windows

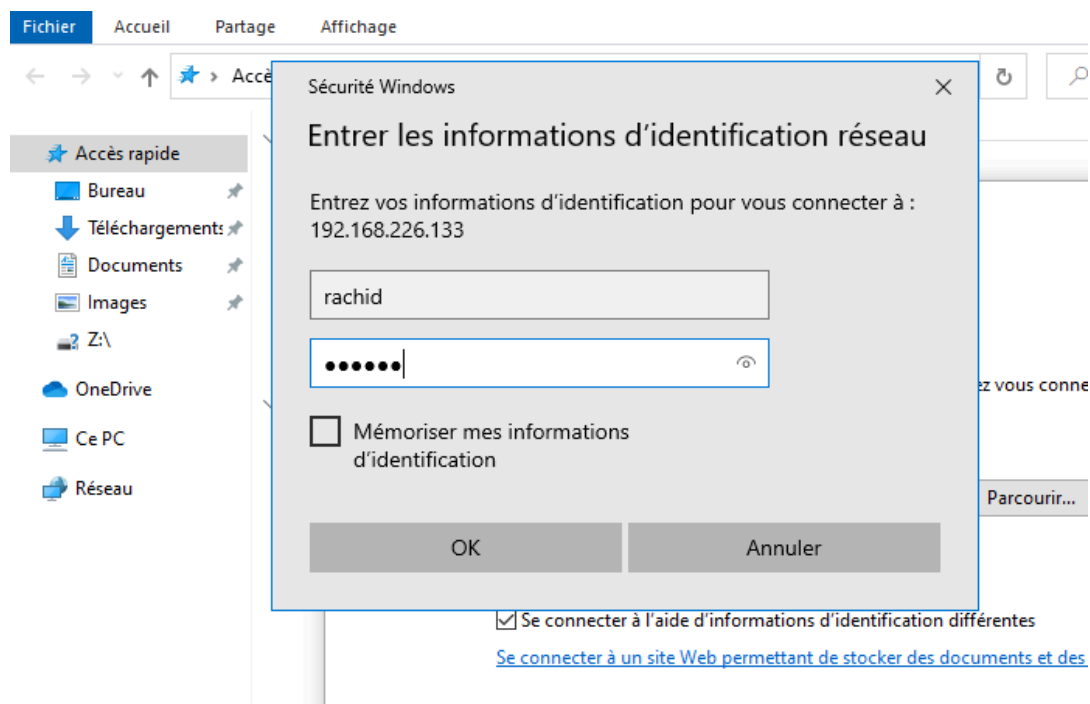
Pour tester l'accès au partage, j'ai pris une machine Windows 10 standard. Pour accéder au partage, il y a plusieurs possibilités : à partir de l'explorateur de fichiers Windows, d'un lecteur réseau, de la commande `net use` voire même `New-PSDrive` en PowerShell.



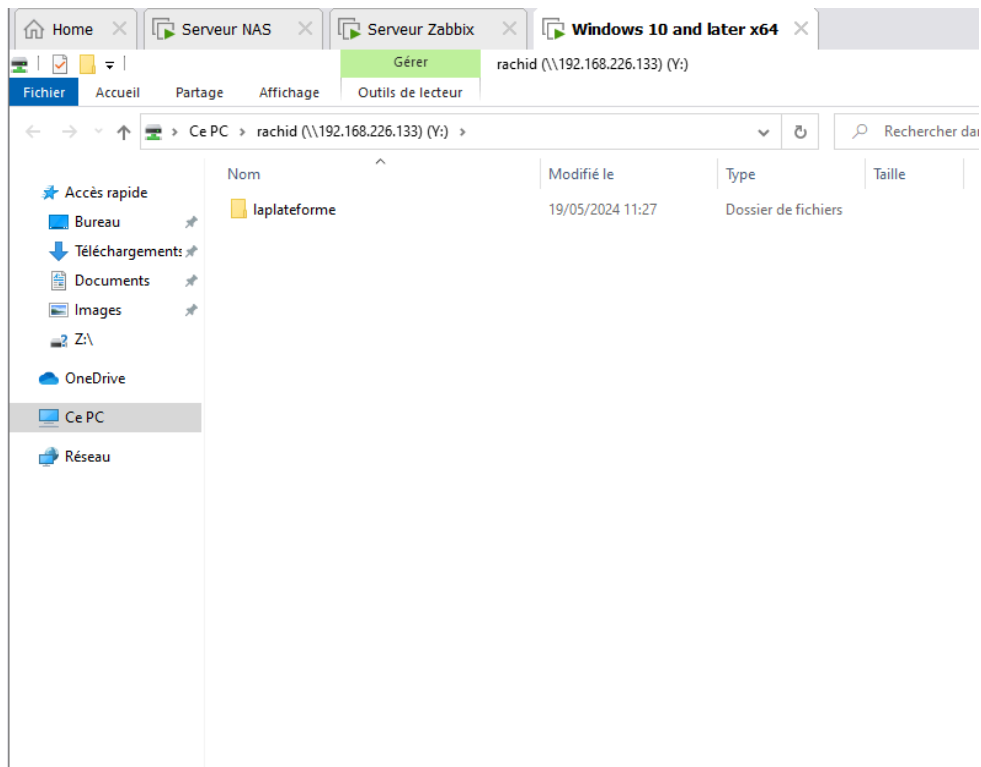
Ici, on spécifie comme dans l'image au-dessus l'adresse IP de votre serveur NAS et le nom du dossier de partage.



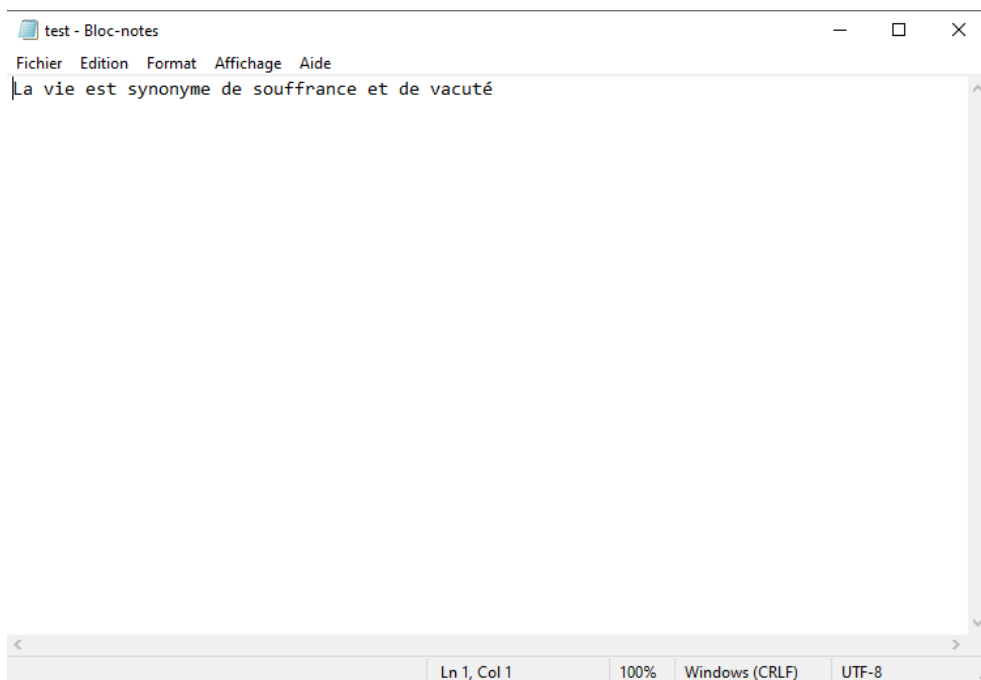
Ensuite on rentre le nom de l'utilisateur qu'on a créé et ajouter à notre groupe auquel le dossier appartient :



**Et voilà, on est dedans, qu'est ce que ça fait du bien !!!!**



Pour pousser un peu plus loin, j'ai créé un fichier test dans le sous-dossier "laplateforme" que je vais essayer d'éditer.



## Accéder au partage Samba depuis une autre machine Linux

Vous pouvez utiliser un outil appelé **smblient** pour accéder à Samba depuis la ligne de commande. Ce package n'est pas inclus par défaut sur la plupart des distributions Linux, vous devrez donc l'installer avec votre gestionnaire de package local.

```
root@debian:~# sudo apt-get install smbclient
```

Après, pour accéder au dossier en se servant du client smb qu'on vient d'installer, on exécute la commande sous le format suivant :

```
root@zabbix-server:~# smbclient //192.168.226.133/rachid -U rachid
```

Ici, Nous avons la commande **smbclient** suivie de **l'adresse IP** de votre serveur Samba et préciser ensuite l'utilisateur qui dans ce cas précis est **"rachid"**

Si Rachid ou un autre utilisateur souhaite accéder au partage commun (**everyone**), modifiez la commande comme suit :

```
root@zabbix-server:~# smbclient //192.168.226.133/everyone -U rachid
```

Après avoir exécuté la smbclient commande, vous serez invité à saisir le mot de passe Samba et connecté à une interface de ligne de commande rappelant l'interface texte FTP :

```
root@zabbix-server:~# smbclient //192.168.226.133/everyone -U rachid
Password for [WORKGROUP\rachid]:
Try "help" to get a list of possible commands.
smb: \>
```

Cette interface est particulièrement utile pour tester les noms d'utilisateur et les mots de passe ainsi que l'accès en lecture-écriture. Par exemple,

vous pouvez créer un répertoire et répertorier son contenu, faire un **ls** pour afficher le contenu de son dossier comme suit :

```
root@zabbix-server:~# smbclient //192.168.226.133/everyone -U rachid
Password for [WORKGROUP\rachid]:
Try "help" to get a list of possible commands.
smb: \> mkdir test
smb: \> ls
.                D          0  Mon May 20 18:33:13 2024
..               D          0  Sun May 19 11:40:20 2024
samba.txt.txt    A        86  Sun May 19 11:55:28 2024
Nouvelle image bitmap.bmp A          0  Sun May 19 11:52:57 2024
public           D          0  Sun May 19 11:53:14 2024
test             D          0  Mon May 20 18:33:13 2024

30769060 blocks of size 1024. 29180664 blocks available
smb: \> rm samba.txt.txt
smb: \>
```

## VI – MISE EN PLACE DE LA VIRTUALISATION

### Installer et Configurer KVM

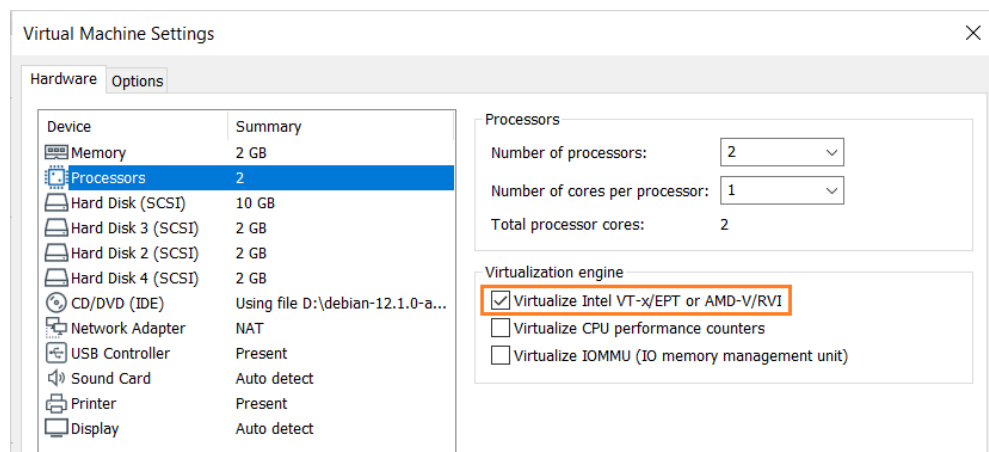
Nous vérifions si notre machine prend en charge la virtualisation :

**grep -E 'vmx|svm' /proc/cpuinfo**

```
allaoui@nas-server:~$ grep -E 'vmx' /proc/cpuinfo
allaoui@nas-server:~$
```

Si cette commande ne renvoie aucun résultat, cela signifie que votre ordinateur ne prend pas en charge la virtualisation.

Heureusement, nous utilisons une machine virtuelle, donc nous pouvons simplement accéder aux paramètres de la VM et cocher la case (voir image) pour activer la virtualisation.



- Maintenant si l'on refait la commande : **grep -E 'vmx|svm' /proc/cpuinfo**

On devrait avoir ça comme résultat :

```

allaoui@nas-server:~$ grep -E 'vmx|svm' /proc/cpuinfo
flags              : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss
syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid tsc_known_freq pni p
clmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervis
or lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid ept_ad fsgsbas
e tsc_adjust bmi1 avx2 smep bmi2 invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec xsaves arat md_clear flush_l1d a
rch_capabilities
vmx flags          : vnmi invvpid ept_x_only ept_ad tsc_offset vtpr mtf ept vpid unrestricted_guest ple ept_mode_based_exec
flags              : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss
clmulqdq vmx ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervis
or lahf_lm abm 3dnowprefetch cpuid_fault invpcid_single pti ssbd ibrs ibpb stibp tpr_shadow vnmi ept vpid ept_ad fsgsbas
e tsc_adjust bmi1 avx2 smep bmi2 invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec xsaves arat md_clear flush_l1d a
rch_capabilities
vmx flags          : vnmi invvpid ept_x_only ept_ad tsc_offset vtpr mtf ept vpid unrestricted_guest ple ept_mode_based_exec
allaoui@nas-server:~$

```

- Mettez à jour les paquets et installez KVM :

**sudo apt-get update**

**sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virtinst -y**

- Chargez les modules KVM : **sudo modprobe kvm**
- Vérifiez que les modules KVM sont correctement chargés :

**lsmod | grep kvm**

```

allaoui@nas-server:~$ sudo modprobe kvm
allaoui@nas-server:~$ lsmod | grep kvm
kvm_intel          380928  0
kvm                1142784  1 kvm_intel
irqbypass         16384   1 kvm
allaoui@nas-server:~$

```

- On vérifie l'existence de /dev/kvm : **ls -l /dev/kvm**

```

allaoui@nas-server:~$ ls -l /dev/kvm
crw-rw---- 1 root kvm 10, 232 May 22 13:07 /dev/kvm
allaoui@nas-server:~$

```

- Ajoutez votre utilisateur au groupe **kvm** : **sudo usermod -aG kvm \$USER**

**newgrp kvm**

```

allaoui@nas-server:~$ sudo usermod -aG kvm $USER
allaoui@nas-server:~$ newgrp kvm
allaoui@nas-server:~$ groups allaoui
allaoui : allaoui cdrom floppy sudo audio dip www-data video plugdev users kvm netdev
allaoui@nas-server:~$

```

Maintenant que KVM est installé sur votre machine Debian, nous allons faire un essai et créer une VM Vide sans OS juste pour tester



## Test de l'installation et création d'une VM

1. Assurez-vous que les services KVM sont en cours d'exécution :

**sudo systemctl status libvirt**

Si le service n'est pas actif, démarrez-le : **sudo systemctl start libvirt**

**sudo systemctl enable libvirt**

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
allaoui@nas-server:~$ sudo systemctl status libvirt
[sudo] password for allaoui:
● libvirt.service - Virtualization daemon
   Loaded: loaded (/lib/systemd/system/libvirt.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-05-22 13:14:11 EDT; 23s ago
     TriggeredBy: ● libvirt-admin.socket
                  ● libvirt.socket
                  ● libvirt-ro.socket
        Docs: man:libvirt(8)
              https://libvirt.org
    Main PID: 673 (libvirt)
      Tasks: 19 (limit: 32768)
     Memory: 33.2M
        CPU: 471ms
    CGroup: /system.slice/libvirt.service
            └─673 /usr/sbin/libvirt --timeout 120

May 22 13:14:11 nas-server systemd[1]: Starting libvirt.service - Virtualization daemon...
May 22 13:14:11 nas-server systemd[1]: Started libvirt.service - Virtualization daemon.
allaoui@nas-server:~$
```

2. Cette commande, **sudo virsh list --all**, permet d'afficher la liste de toutes nos machines virtuelles, même celles qui sont arrêtées.

```
allaoui@nas-server:~$ sudo virsh list --all
 Id   Name   State
-----
allaoui@nas-server:~$
```

Comme vous pouvez le constater, la liste est vide, ce qui est normal vu que nous n'avons pas encore créé de machine virtuelle.

Maintenant, nous allons créer une machine virtuelle. Cependant, en raison de restrictions spécifiques à KVM, il n'est pas possible de créer une VM sans une image d'installation.

Pour contourner cette limitation, nous allons générer un fichier ISO vide en utilisant la commande suivante, que nous utiliserons comme substitut :

```
dd if=/dev/zero of=os_vide.iso bs=1M count=1
```

Cette commande crée un fichier nommé "os\_vide.iso" qui est un ISO vide d'une taille de 1 Mo. Nous pourrions utiliser ce fichier lors de la création de la machine virtuelle

```
allaoui@nas-server:~$ dd if=/dev/zero of=os_vide.iso bs=1M count=1
1+0 records in
1+0 records out
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00152997 s, 685 MB/s
allaoui@nas-server:~$ ls
os_vide.iso
allaoui@nas-server:~$
```

Changez les permissions de l'ISO pour qu'il soit accessible par **libvirt** :

```
sudo chmod +r /home/allaoui/os_vide.iso
```

Créez la machine virtuelle :

```
sudo virt-install \  
--name winxp \  
--memory 1024 \  
--vcpus 2 \  
--network none \  
--graphics none \  
--os-variant winxp \  
--disk size=1 \  
--cdrom /home/allaoui/os_vide.iso
```

```
allaoui@nas-server:~$ sudo virt-install \  
> --name winxp \  
> --memory 1024 \  
> --vcpus 2 \  
> --network none \  
> --graphics none \  
> --os-variant winxp \  
> --disk size=1 \  
> --cdrom /home/allaoui/os_vide.iso  
WARNING: CDROM media does not print to the text console by default, so you likely will not see text install output. You  
might want to use --location. See the man page for examples of using --location with CDROM media  
  
Starting install...  
Allocating 'winxp.qcow2' | 0 B 00:00:00 ...  
Creating domain... | 0 B 00:00:00  
Running text console command: virsh --connect qemu:///system console winxp  
Connected to domain 'winxp'  
Escape character is ^] (Ctrl + J)
```

Appuyez sur Ctrl + C pour arrêter le processus de création.

Maintenant, si nous réexécutons la commande pour afficher toutes nos machines, nous devrions voir que la machine que nous venons de créer est répertoriée et qu'elle est en cours d'exécution : **sudo virsh list --all**

```
allaoui@nas-server:~$ sudo virsh list --all
 Id    Name    State
-----
 1     winxp   running
allaoui@nas-server:~$
```

Vous devriez voir la machine virtuelle "winxp" dans la liste.

Commandes utiles pour gérer la machine virtuelle

Arrêter la machine virtuelle : **sudo virsh shutdown winxp**

Forcer l'arrêt de la machine virtuelle (si elle est en cours d'usage) :

**sudo virsh destroy winxp**

Supprimer la définition de la machine virtuelle :

**sudo virsh undefine winxp**

Supprimer complètement le fichier de disque de la machine virtuelle :

**sudo rm -f /var/lib/libvirt/images/winxp.qcow2**

Voilà, notre environnement de virtualisation KVM est maintenant configuré et opérationnel !

---

## VII – CONFIGURATION DES SAUVEGARDES AVEC RSYNC ET SERVEUR DE SECOURS

### First Step : Installation, Attribution d'Adresse IP, Configuration et Montage du RAID

Nous allons dupliquer le serveur actuel en créant un autre serveur avec la même configuration, disque et RAID afin de sauvegarder les données à l'aide de Rsync.

```
Debian GNU/Linux 12 nas-backup tty1
Hint: Num Lock on

nas-backup login: allaoui
Password:
Linux nas-backup 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu May 23 06:13:56 EDT 2024 from 192.168.1.1 on pts/0
allaoui@nas-backup:~$ _
```

Après avoir terminé l'installation, fixé l'adresse IP, installé et configuré le système, et monté le RAID, nous allons maintenant passer à la mise en place de Rsync.

### Let's do it ! : Installation et transfert !

Nous allons ajouter l'IP et le nom de machine de notre Serveur Backup dans le fichier `/etc/hosts` de notre serveur NAS : **sudo nano /etc/hosts**

```
GNU nano 7.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    nas-server
192.168.1.15 nas-backup
```

Ensuite installer Rsync (sur les deux serveurs) avec la commande :

**sudo apt install rsync**

```
allaoui@nas-server:~$ sudo apt install rsync
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
rsync is already the newest version (3.2.7-1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
allaoui@nas-server:~$
```

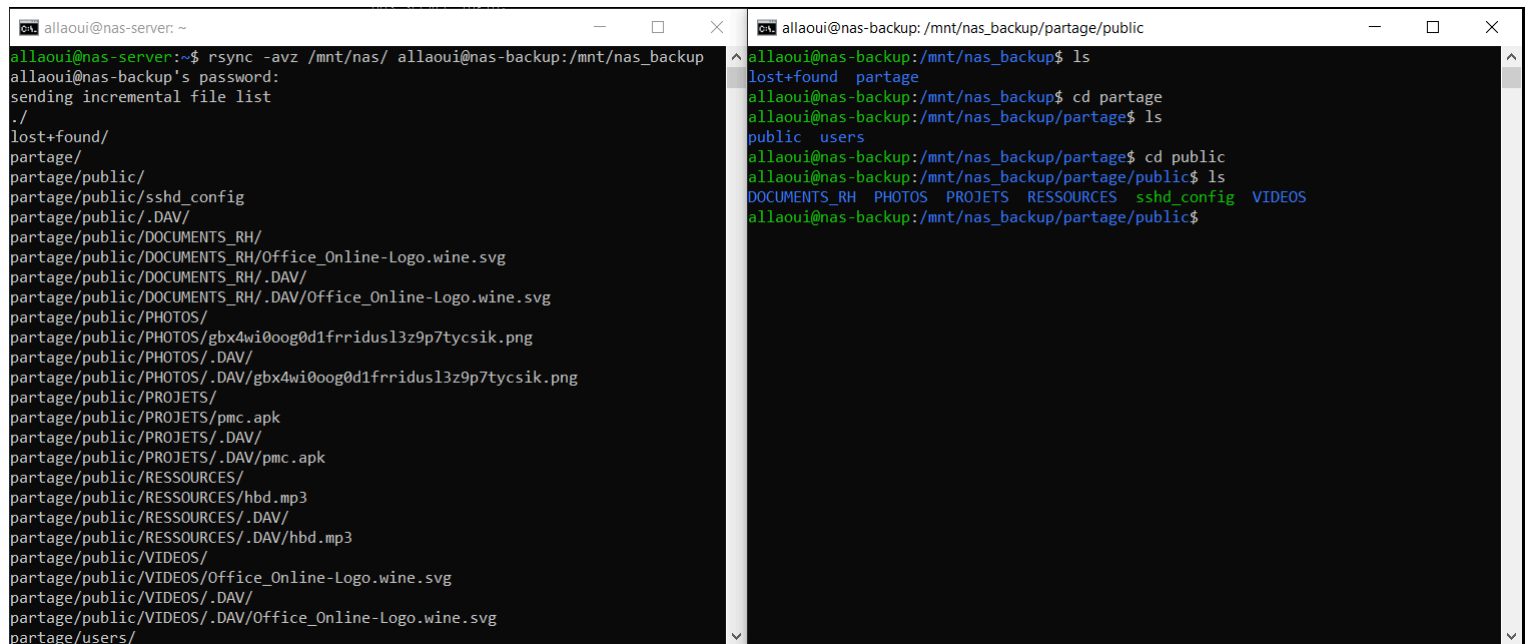
Assurez-vous maintenant que le dossier de partage du **serveur NAS**, ainsi que ses sous-dossiers, ont les droits adéquats. Pour changer les droits, utilisez la commande suivante : **sudo chmod -R 775 /mnt/nas**

Sur le **serveur de sauvegarde** (BACKUP), assurez-vous que le dossier de stockage appartient bien à votre utilisateur et non à root. Utilisez cette commande : **sudo chown -R allaoui:allaoui /mnt/nas\_backup**

```
allaoui@nas-backup:/$ ls -l /mnt/nas_backup/
total 16
drwx----- 2 root root 16384 May 23 06:21 lost+found
allaoui@nas-backup:/$ sudo chown -R allaoui:allaoui /mnt/nas_backup
[sudo] password for allaoui:
allaoui@nas-backup:/$ ls -l /mnt/nas_backup/
total 16
drwx----- 2 allaoui allaoui 16384 May 23 06:21 lost+found
allaoui@nas-backup:/$
```

Maintenant que tout est prêt, nous allons passer à la synchronisation. Pour lancer la synchronisation, exécutez la commande suivante sur le serveur NAS :

**rsync -avz /mnt/nas/ allaoui@nas-backup:/mnt/nas\_backup**



The image shows two terminal windows side-by-side. The left window is titled 'allaoui@nas-server: ~' and shows the execution of the command 'rsync -avz /mnt/nas/ allaoui@nas-backup:/mnt/nas\_backup' with the password prompt. The output shows a list of files being synchronized from the NAS to the backup server. The right window is titled 'allaoui@nas-backup: /mnt/nas\_backup/partage/public' and shows the directory structure of the backup, including 'lost+found', 'partage', and 'public' subdirectories, with 'public' containing various files and subdirectories like 'ssh\_config', 'users', 'DOCUMENTS\_RH', 'PHOTOS', 'PROJETS', 'RESSOURCES', and 'VIDEOS'.

**COMME VOUS POUVEZ LE VOIR SUR LES IMAGES, LE TRANSFERT S'EST BIEN PASSÉ !!**

## Générer une clé SSH

Nous allons maintenant passer à l'automatisation du processus.

Comme vous l'avez peut-être remarqué, la synchronisation avec **Rsync** se fait via **SSH**, et lorsqu'on exécute la commande de transfert, il nous est demandé de nous authentifier avec un mot de passe. Pour l'automatisation, nous ne serons pas toujours présents pour saisir un mot de passe.

Dans cette étape, nous allons générer une clé SSH sur notre serveur NAS et ensuite la copier sur notre serveur de sauvegarde. Cela nous permettra de nous authentifier automatiquement sans avoir à saisir un mot de passe à chaque fois.

Voici les étapes à suivre :

### 1. Générer une clé SSH sur le serveur NAS

Exécutez la commande suivante sur le serveur NAS : **ssh-keygen -t rsa**

Appuyez sur Entrée pour accepter les emplacements de fichiers par défaut et laissez la passphrase vide en appuyant sur Entrée deux fois.

```
allaoui@nas-server:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/allaoui/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/allaoui/.ssh/id_rsa
Your public key has been saved in /home/allaoui/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:09tWulaoC9heHF8xzu80paxA7bp1JHg+oC3axdA4yI4 allaoui@nas-server
The key's randomart image is:
+----[RSA 3072]-----+
|
|               o
|      . . o o . o o
|     o + S * o.= .
|    o  O B O.=.o.
|   E . + * B.O.o.o
|     o + o.B.o o.
|    . . . ++o .
+----[SHA256]-----+
allaoui@nas-server:~$
```

## 2. Copier la clé publique sur le serveur de sauvegarde

Utilisez la commande suivante pour copier la clé publique sur le serveur de sauvegarde : **ssh-copy-id nas-backup**

```
allaoui@nas-server:~$ ssh-copy-id nas-backup
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/allaoui/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
allaoui@nas-backup's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'nas-backup'"
and check to make sure that only the key(s) you wanted were added.
allaoui@nas-server:~$
```

Vous serez invité à entrer le mot de passe du serveur de Sauvegarde.

## 3. Vérifier la connexion SSH sans mot de passe

Testez la connexion SSH pour vous assurer qu'elle fonctionne sans demander de mot de passe : **ssh allaoui@nas-backup**

```
allaoui@nas-server:~$ ssh allaoui@nas-backup
Linux nas-backup 6.1.0-21-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.90-1 (2024-05-03) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Thu May 23 08:39:29 2024 from 192.168.1.14
allaoui@nas-backup:~$
```

Si tout fonctionne correctement, vous devriez vous connecter sans être invité à entrer un mot de passe.

## Automatisation de Rsync

Maintenant que l'authentification sans mot de passe est en place, nous pouvons automatiser la synchronisation avec **Rsync** en ajoutant une tâche cron.

Ouvrez le fichier de crontab pour l'édition : **crontab -e**

On vous demandera de choisir un éditeur faite **1** pour **nano**

```
allaoui@nas-backup:~$ crontab -e
no crontab for allaoui - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
```

Ensuite on va ajouter notre ligne de commande et aussi définir l'intervalle de synchronisation :

**\* \* \* \* \* rsync -avz /mnt/nas/ allaoui@nas-backup:/mnt/nas\_backup**

Cette commande permet de faire une synchronisation toute les 1 minutes

```
# m h dom mon dow  command
* * * * * rsync -avz /mnt/nas/ allaoui@nas-backup:/mnt/nas_backup
```

Enregistrez et fermez le fichier de crontab.

Avec ces étapes, la synchronisation avec **Rsync** est maintenant automatisée et ne nécessite pas de saisie de mot de passe.



## VIII – INSTALLATION D'UNE INTERFACE DE GESTION

### Installation de Webmin

1. Installer curl : **sudo apt install curl**
2. Télécharger le script de configuration Webmin : **curl -o setup-repos.sh https://raw.githubusercontent.com/webmin/webmin/master/setup-repos.sh**
3. Rendre le script exécutable et l'exécuter :  
**sudo chmod +x setup-repos.sh**  
**sudo ./setup-repos.sh**
4. Installer Webmin avec les recommandations : **sudo apt-get install --install-recommends webmin**
5. Confirmer l'installation en répondant "y" pour dire yes lorsque demandé.
6. Accéder à l'interface Webmin :

Ouvrez votre navigateur et entrez l'adresse IP du serveur suivie du port **10000**.

Par exemple : **http://192.168.1.14:10000**

7. Connexion à Webmin : Utilisez un compte avec des privilèges **sudo**, soit le compte **root**, soit un autre compte avec les droits nécessaires.

